Constraint Grammars

Thierry Martinez

Acknowledgments to Rémy Haemmerlé for the original impulse to this work.

Contraintes Project-Team, INRIA Paris-Rocquencourt, France

21 July 2008
Outline

1. Introduction.


4. Conclusions & Perspectives.
Constraint Sets.

Observational confluence

\[
\begin{align*}
\text{nil}(L) \setminus \text{positive_list}(L) & \iff \text{true}. \\
\text{list}(L, H, T) \setminus \text{positive_list}(L) & \iff H > 0, \text{positive_list}(T).
\end{align*}
\]

- Confluent for queries modelling a list (first arguments of list tokens are pair-wise distincts).
- Non-confluent for queries modelling a tree (with two list tokens sharing a common first argument).

Characterizing (L)CC Stores

Describing Global Constraints Given a set of constraints \(S\):

- either there exists a satisfied constraint in \(S\):

\[
\text{REGULAR}(x_1, \ldots, x_n, L) \text{ with } S = \{x_1 = u_1 \land \cdots \land x_n = u_n\}_{u \in L}
\]

- or all the constraints of \(S\) are satisfied:

\[
\text{ALL-DIFFERENT}(x_1, \ldots, x_n) \text{ with } S = \{x_i \neq x_j\}_{i \neq j}
\]
Linear Constraint Systems.

Let $V$ be a set of variables.

- If $C$ is a set of formulae over the set of variables $V$, let $\overline{C}$ be the closure of $C$ by conjunction ($\otimes$) and variable hiding ($\exists$).
- If $\vdash$ is a binary relation over $C$, let $\widehat{\vdash}$ be the closure of $\vdash$ by the rules of linear logic.

**Definition**

A (linear) **constraint system** is a pair $(C, \vdash)$ where:

- $C$ is a set of formulae closed by conjunction ($\otimes$) and variable hiding ($\exists$);
- $\vdash \subseteq C^2$ is a relation closed by the rules of linear logic.
Examples of Constraint Systems

Let $\Sigma$ be a signature. Let $t(\Sigma) \doteq \{ f(v_1, \ldots, v_n) | f/ n \in \Sigma \}$.

**linear-token system**

The **linear-token system** over $\Sigma$ is the constraint system $\text{TOK}(\Sigma) \doteq (\mathbb{C}[t(\Sigma)], \hat{\emptyset})$.

**linear-token system with equality**

The **linear-token system with equality** over $\Sigma$ is the constraint system $\text{TOK}_=(\Sigma) \doteq (\mathbb{C}[t(\Sigma) \cup \{ !(x = y | x, y \in V) \}], \hat{\mathbb{I}})$, where $\hat{\mathbb{I}}$ is the smallest relation with the axioms of the equality theory:

- $\mathbb{I}!(x = x)$;
- $!(x = y) \mathbb{I}!(y = x)$;
- $!(x = y) \otimes !(y = z) \mathbb{I}!(x = z)$;
- $f(x_0, \ldots, x_n) \otimes !(x_0 = y_0) \otimes \cdots \otimes !(x_n = y_n) \mathbb{I} f(y_0, \ldots, y_n)$
Context-Free Constraint Grammars (CFCG).

Definition

A context-free constraint grammar (CFCG) is a tuple \((V, C, \vdash, \Sigma, P)\) where:

- \(V\) is a set of variables and \((C, \vdash)\) is a constraint-system over \(V\);
- \(\Sigma\) is the signature of non-terminal symbols:
  \[ N \equiv \{ f(y_1, \ldots, y_n) | f / n \in \Sigma \text{ and } y_1, \ldots, y_n \in V \text{ pair-wise distincts} \} \]
- \(P \subseteq N \times \overline{C}\) is the set of productions, with \(\overline{C} \equiv \overline{C}[C \cup N]\).
  Every production \((f(y_1, \ldots, y_n), u) \in P\) is such that \(fv(u) \subseteq \{y_1, \ldots, y_n\}\).
  A production is denoted: \(f(y_1, \ldots, y_n) \colon= u\).

Example:

\[
\begin{align*}
h(X, Y) & \colon= \text{edge}(X, Y) \\
h(X, Y) & \colon= \exists Z. \text{edge}(X, Z) \otimes h(Z, Y)
\end{align*}
\]
Derivations for CFCGs.

Let \((V, C, \vdash, \Sigma, P)\) be a context-free constraint grammar. Let \(\overline{C} \cong C[C \cup N]\).

**Definition**

Let \(\rightarrow \subseteq \overline{C}^2\) be the smallest relation satisfying the following rules:

\[ \begin{align*}
\frac{f(y_1, \ldots, y_n) \equiv u}{(f(y_1, \ldots, y_n))\sigma \rightarrow (u)\sigma} & \quad \frac{u \rightarrow v}{\exists x. u \rightarrow \exists x v} & \quad \frac{u \rightarrow v}{w \otimes u \rightarrow w \otimes v}
\end{align*} \]

A derivation is a sequence \((c_i)_i\) with elements in \(\overline{C}\) such that \(c_0 \rightarrow c_1 \rightarrow \ldots\).

**Example:**

\[ h(X, Y) \equiv \text{edge}(X, Y) \]

\[ h(X, Y) \equiv \exists Z. \text{edge}(X, Z) \otimes h(Z, Y) \]

\[ h(A, B) \rightarrow \exists X_1. \text{edge}(A, X_1) \otimes h(X_1, B) \]

\[ \rightarrow \exists X_1 X_2. \text{edge}(A, X_1) \otimes \text{edge}(X_1, X_2) \otimes h(X_2, B) \]

\[ \rightarrow \exists X_1 X_2. \text{edge}(A, X_1) \otimes \text{edge}(X_1, X_2) \otimes \text{edge}(X_2, B) \]
Languages accepted by CFCGs.

Let $G = (V, C, \vdash, \Sigma, P)$ be a context-free constraint grammar. Let $\overline{C} = C[C \cup N]$.

**Definition**

The **constraint language** accepted by a schema $c_0 \in \overline{C}$ and the grammar $G$ is:

$$L_G(c_0) = \{ c \in C | (\exists n)(\exists (c_i)_{0 \leq i \leq n})(c_0 \rightarrow \cdots \rightarrow c_n \wedge c_n \vdash c) \}$$

**Example:**

$$h(X, Y) ::= \text{edge}(X, Y)$$

$$h(X, Y) ::= \exists Z. \text{edge}(X, Z) \otimes h(Z, Y)$$

$$L_G(h(A, B)) = \{ \text{hamiltonian paths from } A \text{ to } B \}$$
Claim
Over linear token systems, context-free constraint grammars have the same expressive power than hyperedge replacement graph grammars.


Theorem (K.-J. Lange and E. Welzl 87)
Hyperedge replacement graph language membership problem is NP-complete.
Hypergraphs.

A labelled set $\hat{E}$ is a tuple $(E, E^\Sigma, E^\lambda)$, with $E^\Sigma$ alphabet and $E^\lambda : E \to E^\Sigma$.

**Definition**

A hypergraph is a tuple $(V, \hat{E}, S, \rightarrow)$, where:

- $V$ is the set of vertices; $\hat{E}$ is the labelled set of hyperedges;
- $S$ is the alphabet of selectors;
- $\rightarrow \subseteq E \times S \times V$ is the incidence relation. $(e, s, v) \in \rightarrow$ is denoted $e \overset{s}{\rightarrow} v$.
  - $\rightarrow$ is functional: $(\forall e s v_1 v_2) e \overset{s}{\rightarrow} v_1 \land e \overset{s}{\rightarrow} v_2 \Rightarrow v_1 = v_2$

Vertices are represented as fat dots, hyperedges as boxes.

Let $\mathcal{H}(V, E, S)$ the set of hypergraphs over $V, E, S$. 
Embedding of Hypergraphs into Constraints.

Let $H = (V, \dot{E}, S, \rightarrow)$ be a hypergraph.

- For all $e \in E$, let $\text{dom}(e) \doteq \{ s \in S | (\exists v \in V)(e \overset{s}{\rightarrow} v) \}$.
- Let $n(e) \doteq |\text{dom}(e)|$ and $r_e : \{1, \ldots, n(e) \} \rightarrow \text{dom}(e)$.
- Let $\Sigma \doteq \{ f / n \in E^\Sigma \times \mathbb{N} | (\exists e \in E)(f = E^\lambda(e) \text{ and } n = n(e)) \}$.

Definition

The constraint-embedding of the hypergraph $h$ is the following constraint of $\text{TOK}(\Sigma)$:

$$[[h]] \doteq (\exists v_1 \ldots v_n) \left( \bigotimes_{e \in E \land n = n(e) \land y_i = r_e(i)} [E^\lambda(e)](y_1, \ldots, y_n) \right)$$

Proposition

For every closed constraint $c \in \text{TOK}(\Sigma)$, there exists a hypergraph $h$ such that

$$[[h]] ::= c$$
Hypergraph Isomorphism.

**Definition**

Two hypergraphs $h_0 \doteq (V,(E,E^\Sigma,E^\lambda_0),S,\to_0)$ and $h_1 \doteq (V,(E,E^\Sigma,E^\lambda_1),S,\to_1)$ are **isomorphic**, denoted $h_0 \simeq h_1$, when there exist $\sigma_V : V \leftrightarrow V$ and $\sigma_E : E \leftrightarrow E$ such that:

- $(\forall e \in E)(E^\lambda_0(e) = E^\lambda_1(\sigma_E(e)))$;
- $(\forall e \in E)(\forall s \in S)(\forall v \in V)(e \to^s_0 v \iff (\sigma_E(e) \to^s_1 \sigma_V(v)))$.

**Proposition**

$h_0 \simeq h_1 \iff [h_0] \vdash \vdash [h_1]$
Hyperedge Replacement Graph Grammars (HRGs).

Let $N, V, E, E^\Sigma, S$ be defined accordingly to the following definition.

We denote $\mathcal{H}^+ (N, V, E, E^\Sigma, S)$ the set of augmented hypergraphs, that is hypergraphs whose edges are labelled with $\hat{E}^\Sigma$ and vertices are in $\hat{V}$.

**Definition**

A **h-edge replacement graph grammar** is a tuple $(N, V, E, E^\Sigma, S, P)$, with:

- $N$ is the alphabet of non-terminals, such that $E^\Sigma \cap N = \emptyset$; $\hat{E}^\Sigma \doteq E^\Sigma \cup N$;
- $V$ is the set of vertices;
- $E$ is the set of hyperedges and $E^\Sigma$ the set of ground labels;
- $S$ is the alphabet of selectors, such that $V \cap S = \emptyset$; $\hat{V} \doteq V \cup S$;
- $P \subseteq N \times \mathcal{P}(S) \times \mathcal{H}^+ (N, V, E, E^\Sigma, S)$ is the set of productions.
Derivations for HRGs.

Let \((N, V, E, E^\Sigma, S, P)\) be a hyperedge replacement graph grammar.

**Definition**

Let \(\rightarrow \subseteq H^+(N, V, E, E^\Sigma, S)^2\) be the relation such that: 
\((V_0, E_0, S, \rightarrow_0) \rightarrow (V_1, E_1, S, \rightarrow_1)\) iff there exist:

- a hyperedge \(e \in E_0\); a production \((n, S_p, (V_p, E_p, S, \rightarrow_p)) \in P\);
- \(\sigma_{0\to1}^E : E_0 \setminus \{e\} \rightarrow E_1, \sigma_{0\to1}^V : V_0 \rightarrow V_1, \sigma_{p\to1}^E : E_p \rightarrow E_1, \sigma_{p\to1}^V : V_p \setminus S \rightarrow V_1\) such that:
  - \(E_0^\lambda(e) = n; \text{im}(\sigma_{0\to1}^E) \cup \text{im}(\sigma_{p\to1}^E) = E_1\) and \(\text{im}(\sigma_{0\to1}^V) \cup \text{im}(\sigma_{p\to1}^V) = V_1\);
  - \((\forall e_0 \in E_0 \setminus \{e\})(\forall e_p \in E_p)(\forall s s' \in S)(\forall v_0 \in V_0)(\forall v_p \in V_p \setminus S)\)

\[
\begin{align*}
E_0^\lambda(e_0) &= E_1^\lambda(\sigma_{0\to1}^E(e_0)) \\
E_p^\lambda(e_p) &= E_1^\lambda(\sigma_{p\to1}^E(e_p))
\end{align*}
\]

\[
\begin{align*}
e_0 \stackrel{s_0}{\rightarrow} v_0 &\iff \sigma_{0\to1}^E(e_0) \stackrel{s_1}{\rightarrow} \sigma_{0\to1}^V(v_0) \\
e_p \stackrel{s_p}{\rightarrow} v_p &\iff \sigma_{p\to1}^E(e_p) \stackrel{s_1}{\rightarrow} \sigma_{p\to1}^V(v_p) \\
e \stackrel{s_0}{\rightarrow} v_0 \wedge e_p \stackrel{s'_{p}}{\rightarrow} s &\iff \sigma_{p\to1}^E(e_p) \stackrel{s'_{1}}{\rightarrow} \sigma_{0\to1}^V(v_0)
\end{align*}
\]
Embedding of HRGs into CFCGs.

Let $G = (N, V, E, E_\Sigma, S, P)$ be a hyperedge replacement graph grammar.

1. $\cdot f$ can be extended to an embedding of augmented hypergraphs into a scheme of C, with $N = \{ f(y_1, \ldots, y_n) \mid (f, \{y_1, \ldots, y_n\}, h) \in P \}$;

2. therefore $\cdot f$ can be extended to an embedding of HRG productions into context-free constraint productions;

3. therefore $\cdot f$ can be extended to an embedding of HRG grammars into context-free constraint grammars.

**Definition**

The hypergraph language accepted by an augmented hypergraph $h_0 \in H^+(N, V, E, E_\Sigma, S)$ and the grammar $G$ is:

$$L_G(h_0) = \{ h \in H(V, E, S) \mid (\exists n)(\exists (h_i)_{0 \leq i \leq n})(h_0 \to \cdots \to h_n \land h \equiv h_n) \}$$

**Proposition**

For all $h_0 \in H^+(N, V, E, E_\Sigma, S)$, we have $L_G(\cdot f) = L'_G(\cdot f(\cdot f))$. 

Thierry Martinez (INRIA) Constraint Grammars 2008/07/21 15 / 22
With Equality Theory.

**Lemma:** for all constraint $c \in \text{TOK}_=(\Sigma)$, there exists a constraint $c' \in \text{TOK}(\Sigma) \subseteq \text{TOK}_=(\Sigma)$ such that $c \vdash \vdash c'$.

**Proposition**

*For every closed constraint $c \in \text{TOK}_=(\Sigma)$, there exists a hypergraph $h$ such that* 

$$[[h]] \vdash \vdash c$$

**Lemma:** For a schema $c \in \text{TOK}_=(\Sigma)$ and a grammar $G$ over $\text{TOK}_=(\Sigma)$, there exists $c' \in \text{TOK}(\Sigma)$ and a $G'$ over $\text{TOK}(\Sigma)$ such that:

$$L_G(c) = \{ h \in \text{TOK}_=(\Sigma) | \exists h' \in L_{G'}(c'), h \vdash \vdash h' \}$$

**Proposition**

*Over $\text{TOK}_=(\Sigma)$, context-free constraint language membership problem is NP-complete.*
Applications to Global Constraints.

Let $A$ be an alphabet. Every $u = u_1 \cdots u_n \in A^*$ can be represented as a string-hypergraph:

\[
\begin{array}{c}
\bullet &  \overset{0}{\bullet} & u_1 & 1 & \cdots & 0 & u_n & 1 & \bullet \\
\end{array}
\]

Proposition (Habel and Kreowski 87)

*Every context-free string language $G$ can be recognized by a hyperedge replacement graph grammar $\mathcal{G}$.]*

Corollary: Every context-free string language $G$ can be recognized by a context-free constraint grammar $\mathcal{G}$.

Proposition

\[
\text{CONTEXT-FREE}(x_1, \ldots, x_n, G) \equiv (\exists)\mathcal{G}(x_1, \ldots, x_n)
\]
Limitations.

**Definition**

A **jungle** is a hypergraph \((V, \dot{E}, \{\text{in}_k, \text{out}\}, \rightarrow)\) such that:

- \((\forall v \in V)\left(\left|\{e \in E| e^{\text{out}} \rightarrow v\}\right| \leq 1\right)\);
- \(\rightarrow\) is acyclic.


"**Corollary 2:** CHR folding is terminating and confluent."

*Only true if the query is a jungle!*

**Proposition**

\(\{h \in \mathcal{H}(V,E,S)| h \text{ is a jungle}\} \text{ is not a hyperedge replacement graph language.}\)
A two-sorted signature $\Sigma$ is a subset of $F \times \mathbb{N} \times \mathbb{N}$, where $F$ is a countable symbol set. Items $(f, n, m) \in \Sigma$ are denoted $f/(n, m)$.

Let $V$ and $W$ be two set of variables and $L$ a set of labels. The set constraint system is the smallest classical constraint system $(S, \vdash_S)$ such that:

\[
\begin{align*}
\frac{w \in W}{w = \emptyset \in S} & \quad \frac{v \in V \quad w \in W}{w = \{v\} \in S} & \quad \frac{v \in V \quad w \in W}{(v \in w) \in S} & \quad \frac{w \in W \quad l \in L}{w = f\forall l \in S} \\
\frac{w_1 \in W \quad w_2 \in W \quad w_3 \in W}{w_1 = w_2 \; op \; w_3 \in S} & \quad op \in \{\cap, \cup, \setminus\}
\end{align*}
\]

with $w = \{v\} \vdash_S v \in w \quad w = \emptyset \vdash_S v \notin w$

and $v \in w_1 \; op_1 \; v \in w_2 \vdash_S v \in w_1 \; op_2 \; w_2$ with $(op_1, op_2) \in \{ (\land, \cap); (\lor, \cup); (\land \neg, \cap) \} \ldots$. 
Definition

A **domain-sensitive constraint grammar** (DSCG) is a tuple $(V, W, L, C, \vdash, \Sigma, P)$ where:

- $V, W$ are sets of variables, $L$ a set of labels;
- $(C, \vdash)$ is a constraint-system over $V$;
- $\Sigma$ is the two-sorted signature of non-terminal symbols:
  $$N \doteq \left\{ l: f(y_1, \ldots, y_n, Y_1, \ldots, Y_m) | l \in L, f/(n, m) \in \Sigma \text{ and } y_1, \ldots, y_n \in V, Y_1, \ldots, Y_m \in W \right\} \text{ p.-w. dist.}$$
- $P \subseteq N \times \overline{C} \times S$ is the set of productions, with $\overline{C} \doteq \mathbb{C}[C \cup N]$.

A production is denoted: $f(y_1, \ldots, y_n, Y_1, \ldots, Y_m) ::= u|g$. 

Thierry Martinez (INRIA)
Applications of DSCGs.

**Jungles** Let $\Sigma$ be a signature. For all $f/n \in \Sigma$:

$$j(r) ::= \text{eq}(r, f(x_1, \ldots, x_n)) \otimes l_1 : j(x_1) \otimes \cdots \otimes l_n : j(x_n) | r \not\in \text{fv}(l_1) \cup \cdots \cup \text{fv}(l_n)$$

**all-different**

$$a(S) ::= x_1 \neq x_2 | x_1 \in S \land x_2 \in S \setminus \{x_1\}$$

$$\text{ALL-DIFFERENT}(x_1, \ldots x_n) ::= (\forall) a(\{x_1, \ldots, x_n\})$$
Conclusions & Perspectives.

Objective

- Describing set of constraints;
- for observational confluence;
- for defining global constraints.

Contributions

- Definition of constraint grammar;
- as expressive as hyperedge-replacement graph grammar;
- much more easy to define;
- more general.

Future work

- Decidability and complexity analysis for other constraint systems.